Современная Front-End разработка

AngularJS: Сервисы

Николай Зотеев
zoteev.nikolay@simbirsoft.com

План лекции

1. Что такое Сервисы AngularJS?
2. Типы Сервисов
3. Нативные Сервисы AngularJS
4. Promise
5. Мастер-класс

Дополнительные соглашения

```
var app = angular.module("angularjs-learning", []);

app

  // Конфигурирование приложения и сервисов
  .config(["$stateProvider", function ($stateProvider) { ... }])

  // Старт приложения
  .run(["$rootScope", function ($rootScope) { ... }])
```

Названия сервисов начинаем с символа $
Например, $rootScope или $stateProvider

SIMBIRSOFT

ANGULARJS

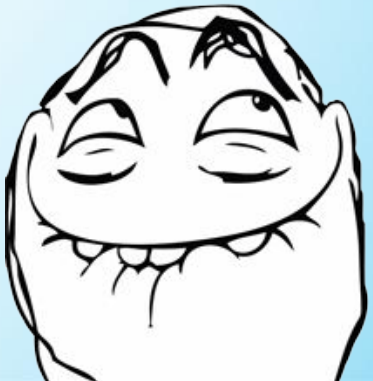
by Google

Сервисы AngularJS представляют специальные объекты или функции, выполняющие некоторые общие для всего приложения задачи. Имейте в виду, что сервисы, не зависимо от типа, это всегда **синглтоны**.

Примечание: Синглтон это шаблон проектирования, который ограничивает тип таким образом, что у него может быть только один экземпляр. Именно с этим экземпляром и ведется работа везде, где он используется.

Для чего используются?

1. Логика (бизнес логика) приложения
2. Запросы к серверу
3. Хранение данных и (или) состояния
4. Коммуникация между компонентами



Типы сервисов

- Constant
- Value
- Factory
- Service
- Provider
- Decorator

Типы сервисов

```
app.constant("$sampleService1", 10);

app.constant("$sampleService2", "sampleService2");

app.constant("$sampleService3", { a: 10 });
```

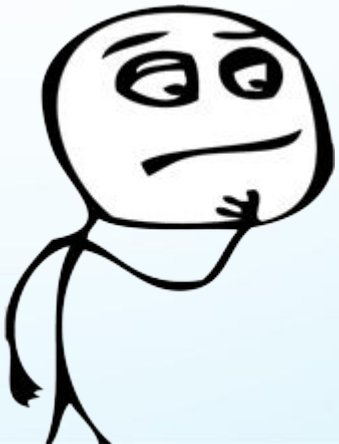
Типы сервисов

```
app.value("$sampleService1", 10);

app.value("$sampleService2", "sampleService2");

    app.value("$sampleService3", { a: 10 });

        app.value("$sampleService4", function (a) {
            return a * a;
        });
```

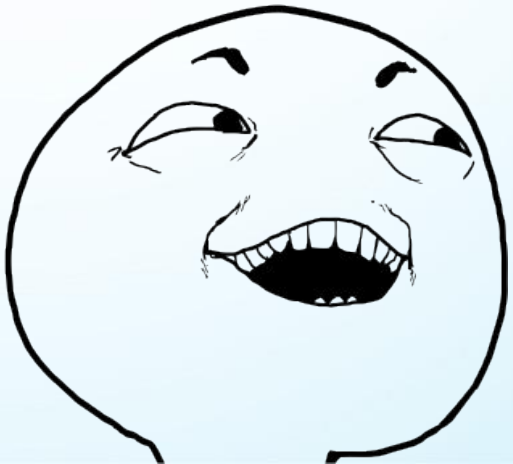


Типы сервисов

```
app.factory("$sampleService", function () {
    var x = 100;
    return {
        a: 20,
        b: function (c) {
            return x * c;
        }
    };
});
```

Типы сервисов

```
app.service("$sampleService", function () {
    this.a = 20;
    this.b = function (c) {
        return Math.round(c);
    };
});
```



Типы сервисов

```
app.provider("$sampleService", function () {
    this.a = 20;
    this.$get = ["$q", function ($q) {
        return {
            b: function (c) {
                return $q.when(c);
            }
        };
    }];
});
```

Типы сервисов

```
app.config(function($provide) {
  $provide.decorator("$sampleService", function($delegate) {
      $delegate.d = function() {
         return "decorated!";
      };
      return $delegate;
  });
});
```

Нативные сервисы

$window – ссылка на глобальный объект window;

$document – jQLite (jQuery) обертка document;

$location – взаимодействие с адресной строкой;

$exceptionHandler – сервис эксепшенов;

$animate – создание анимаций;

$log – логирование;

Нативные сервисы

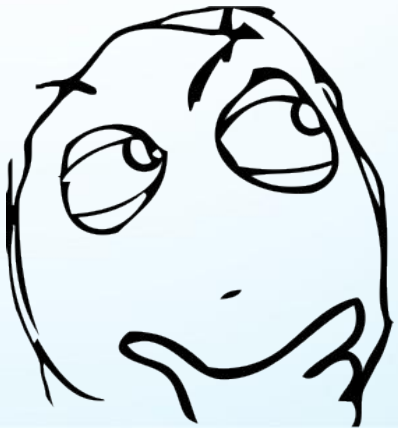
$timeout / $interval – обертки над setTimeout / setInterval;

$q – сервис работы с promise (обещаниями);

$http – HTTP запросы;

$httpBackend – «фейковый» аналог $http,

Используется для Unit тестирования;



Нативные сервисы

$parse – конвертирует Ангулар-выражение в функцию

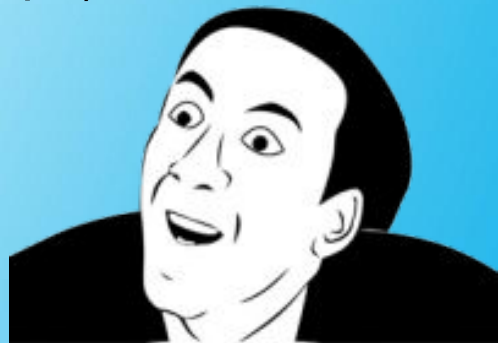
user.name => fn() / getter / setter

$interpolate – «компилирует» текст с разметкой в функцию

Hello, {{user.name}}! => fn(scope) => Hello, Ivan!

$compile – «компилирует» текст или DOM в link функцию.

{{use.name}} => linkFn(scope)



Нативные сервисы

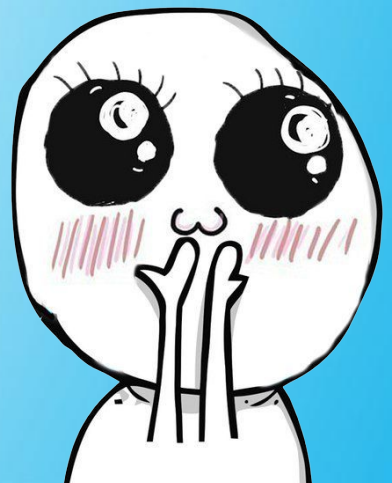
$controller – вызов контроллера;

$filter – создание фильтров:

ng-repeat="friend in friends | orderBy:'age'";

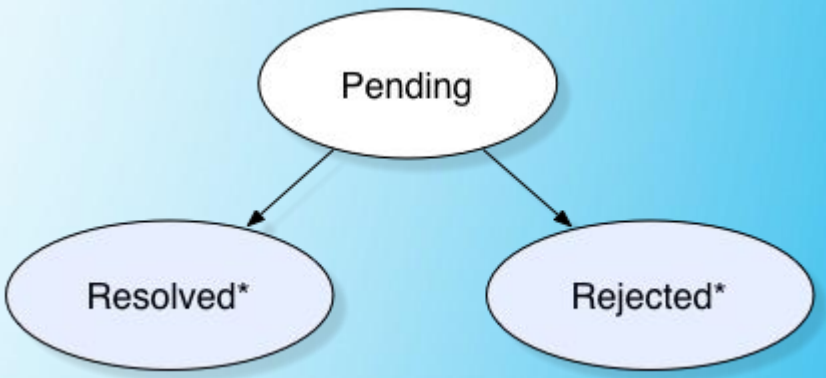
$cacheFactory – создания и получения доступа к кэш-хранилищам;

$templateCache – сервис кеширования шаблонов.



Promise

Promise (Обещание) – специальный объект, который позволяет получить результат выполнения операции отложенный во времени не блокирую очередь выполнения браузера.



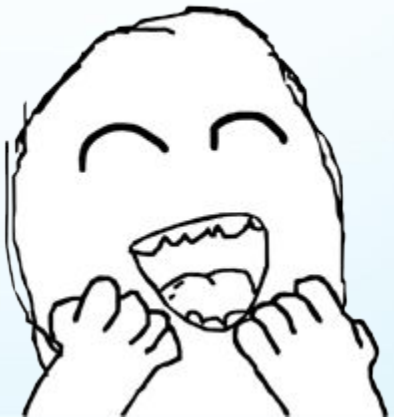
Promise

```
promise.then(
    successCallback,
    errorCallback,
    progressCallbak
);
```

```
promise.then(successCallback);
```

```
promise.catch(errorCallback);
```

```
promise.finally(callback);
```



Promise

```
function asyncGreet(name) {
    var deferred = $q.defer();
    setTimeout(function() {
        deferred.notify("About to greet " + name + ".");
        if (okToGreet(name)) {
            deferred.resolve("Hello, " + name + "! ");
        } else {
            deferred.reject("Greeting " + name + " is not allowed.");
        }
    }, 1000);
    return deferred.promise;
}

var promise = asyncGreet("Robin Hood");
promise.then(function(greeting) {
    alert("Success: " + greeting);
}, function(reason) {
    alert("Failed: " + reason);
}, function(update) {
    alert("Got notification: " + update);
});
```

Что почитать?

- https://habrahabr.ru/post/190342/
- http://stepansuvorov.com/blog/2015/02/с-чего-начать-angularjs-часть3/
- http://stepansuvorov.com/blog/2013/07/встроенные-сервисы-angularjs/
- https://habrahabr.ru/post/189084/
- https://habrahabr.ru/post/221111/

Вопросы?